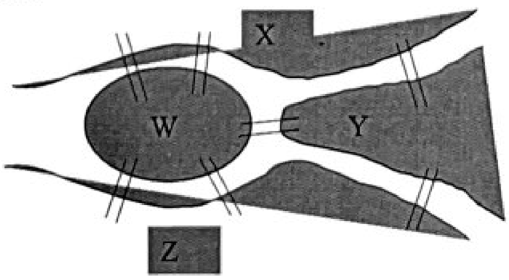
Chapter IV Graph Theory

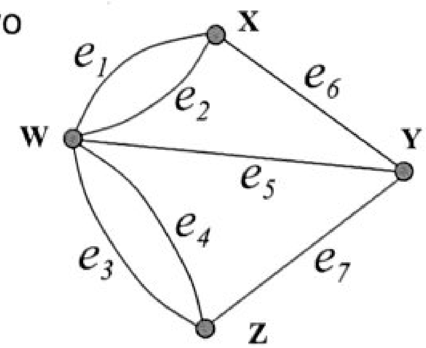
Objectives:

- What Is a Graph?
- Some types of Graphs
- Subgraphs, Bipartite graphs, Connected and Disconnected Graphs
- Connectivity
- Vertex Degree and Counting
- Adjacency and Incidence Matices
- Special Graphs

Graph Models

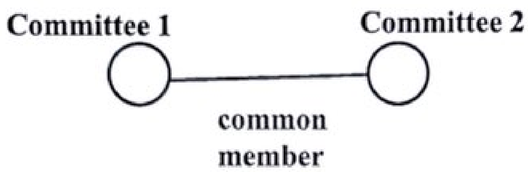
- A **graph** is a set of points, called nodes or vertices, which are interconnected by a set of lines called edges. The study of **graphs**, or **graph** theory is an important part of a number of disciplines in the fields of mathematics, engineering and computer science.
- A *vertex*: a region
- An *edge*: a path(bridge) between two regions





General Model

- ❑ A *vertex*: an object
- ❑ An *edge*: a relation between two objects



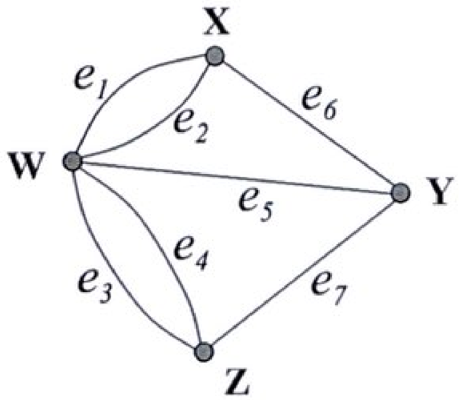
What Is a Graph?

A graph *G* consists of two things:

(i) A set *V*(*G*) whose elements are called vertices, points or nodes.

(ii) A set *E*(*G*) of unordered pairs of distinct vertices called edges.

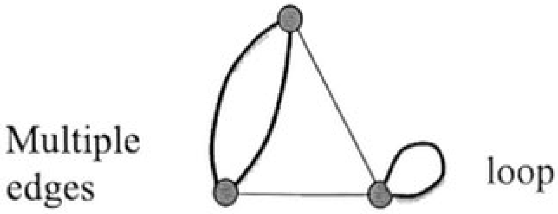
We denote such a graph by G(V, E).



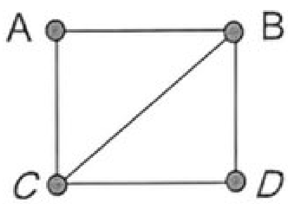
TISHK INTERNATIONAL UNIVERSITY

Types of Edges

- *Loop* : An edge whose endpoints are equal
- *Multiple edges* : Edges have the same pair of endpoints

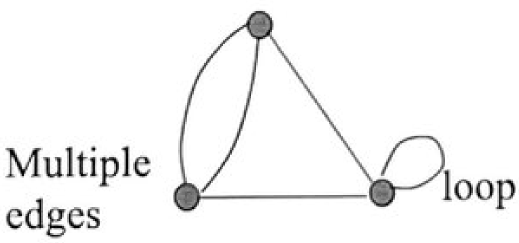


- Two vertices are *adjacent* and are *neighbors* if they are the endpoints of an edge
- Example: *A* and *B* are adjacent
- *while A* and *D* are not adjacent

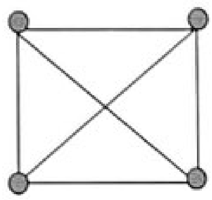


Some Types of Graphs

- *Trivial graph* : the graph with one vertex and no edges, i.e. a single point, is called *Trivial graph*
- *Finite graph* : an graph whose vertex set and edge set are finite
- *Simple graph* : A graph has no loops or multiple edges
- *Multigraph* : is a graph which is permitted to have multiple edges



It is not simple,
it is a multigraph.

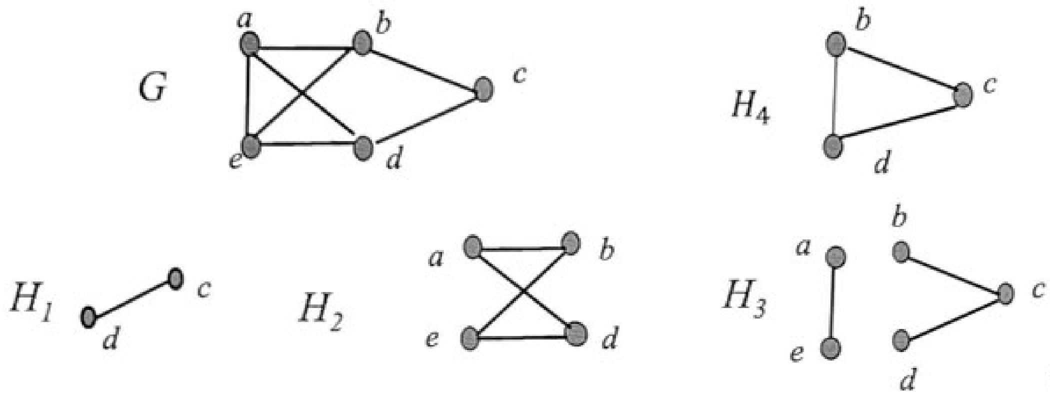


It is a simple graph.

Subgraphs

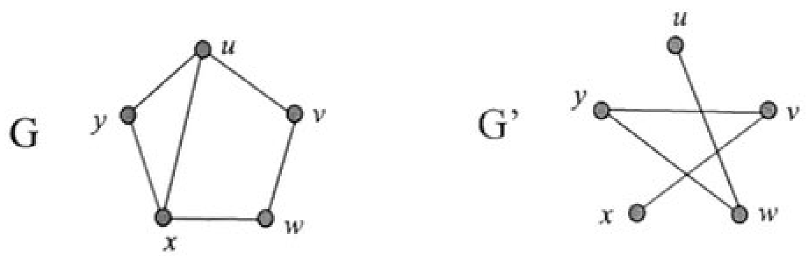
- A ***subgraph*** of a graph G is a graph H such that:
  - $V(H) \subseteq V(G)$ and $E(H) \subseteq E(G)$ and
  - The assignment of endpoints to edges in H is the same as in G.
- Example: H_1, H_2, and H_3 are subgraphs of G, while H_4 is

not a subgraph.



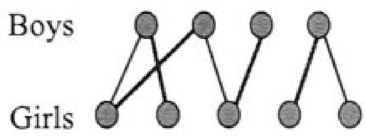
Complement Graph

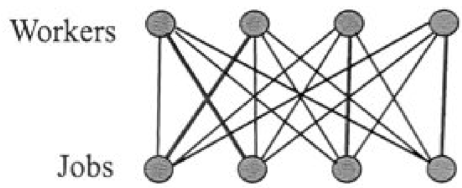
- ***Complement of*** G: The complement G' of a simple graph G:
  - A simple graph
  - $V(G') = V(G)$
  - $E(G') = \{ uv \mid uv \notin E(G) \}$



Bipartite Graphs

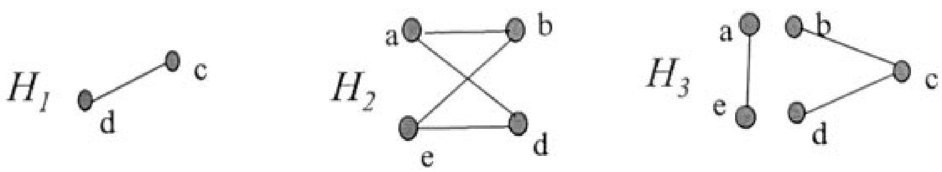
- A graph **G** is **bipartite** if **V(G)** is the union of two disjoint independent sets called **partite sets of G**
- Also: The vertices can be partitioned into two sets such that each set is independent
- Matching Problem
- Job Assignment Problem





Connected and Disconnected Graphs

- ***Connected Graph*** : There exists at least one path between any two of its vertices.
- ***Disconnected*** : Otherwise
- Example:
  - H_1 and H_2 are connected
  - H_3 is disconnected

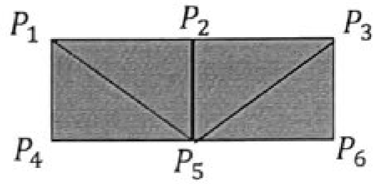


Connectivity

- **Walk:** a list of vertices and edges $v_0, e_1, v_1, e_2, v_2, \ldots, e_k, v_k$ such that, for $1 \le i \le k, the\ edge\ e_i$ has endpoints v_{i-1} and v_i .
- **Trail:** a walk with no repeated edge.
- **Path:** a walk with no repeated vertex; hence a path must be trial.
- **Cycle:** a closed walk such that all vertices are distinct except $v_0 = v_n$. A cycle of length k is called a k-cycle. In a graph, any cycle must have length three or more.

Example. Consider the graph in the following Fig., then

$(P_4, P_1, P_2, P_5, P_1, P_2, P_3, P_6)$ is a walk from P_4 to P_6 . It is not trail since the edge $\{P_1, P_2\}$ is used twice. The sequence $(P_4, P_1, P_5, P_2, P_6)$ is not a walk since there is no edge $\{P_2, P_6\}$. The sequence $(P_4, P_1, P_5, P_3, P_6)$ is a path from P_4 to P_6. The sequence $(P_4, P_1, P_5, P_2, P_3, P_5, P_6)$ is a trail but not path since no edge is used twice and the vertex P_5 is used twice.

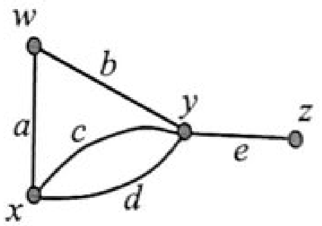


Degree

- If v is an endpoint of an edge e, then we say that e is incident on v. The degree of a vertex v, written deg (v), is equal to the number of edges which are incident on v.
- A vertex is said to be *even* or *odd* according as its degree is an even or an odd number.
- A vertex of degree zero is called an *isolated vertex*.
- An ***even graph*** is a graph with vertex degrees all even.
- The ***length*** of a walk, trail, path, or cycle is its number of edges.

Adjacency matrix

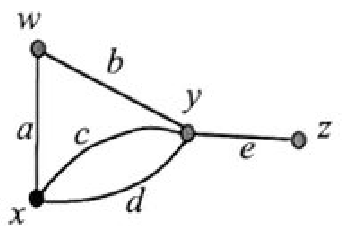
- Let $G = (V, E)$, $|V| = n$ and $|E|=m$
- The ***adjacency matrix*** of G written $A(G)$, is the n-by-n matrix in which entry $a_{i,j}$ is the number of edges in G with endpoints $\{v_i, v_j\}$.



$$
\begin{array}{c} \\ w \\ x \\ y \\ z \end{array}
\begin{array}{c} \begin{array}{cccc} w & x & y & z \end{array} \\
\begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 2 & 0 \\ 1 & 2 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \end{array}
$$

Incidence Matrix

- Let $G = (V, E)$, $|V| = n$ and $|E|=m$
- The ***incidence matrix*** $I(G)$ is the n-by-m matrix in which entry $m_{i,j}$ is 1 if v_i is an endpoint of e_i and otherwise is 0.

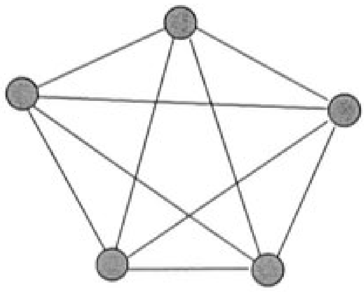


$$
\begin{array}{c} \\ w \\ x \\ y \\ z \end{array}
\begin{array}{c} \begin{array}{ccccc} a & b & c & d & e \end{array} \\
\begin{pmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \end{array}
$$

Special Graphs

- ***Complete Graph*** : a simple graph whose vertices are pairwise adjacent. [A graph is complete if each vertex is connected to every other vertex.]
- The complete graph with n vertices is denoted by Kn
- The degree of each vertex in the complete graph with n vertices equal to (n-1).

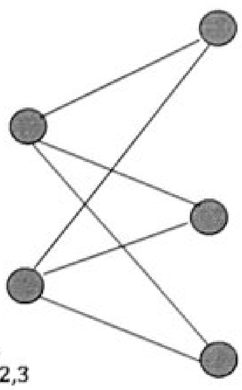
$d(v_i) = n - 1 \; for \; all \; i$



Complete Graph K_5

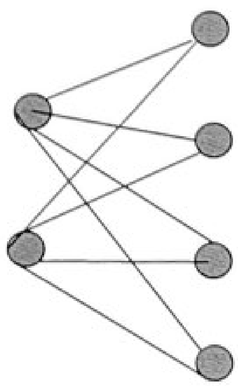
Complete Bipartite Graph

- ***Complete bipartite graph*** (biclique) is a simple bipartite graph such that each vertex of M is connected to each vertex of N, this graph is denoted by Km,n has mn edges, we assume



$K_{2,3}$

Complete Bipartite Graph



$K_{2,4}$

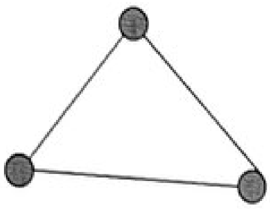
Regular

- ❑ ***Agraph G*** is *regular* of deqree k or *k-reqular* if every vertex has the same degree ***k***.

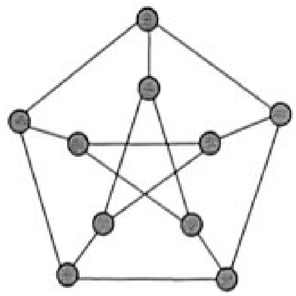
Examples.

0-regular with one vertix

1-regular with two vertices



2-regular with three vertices



3-regular

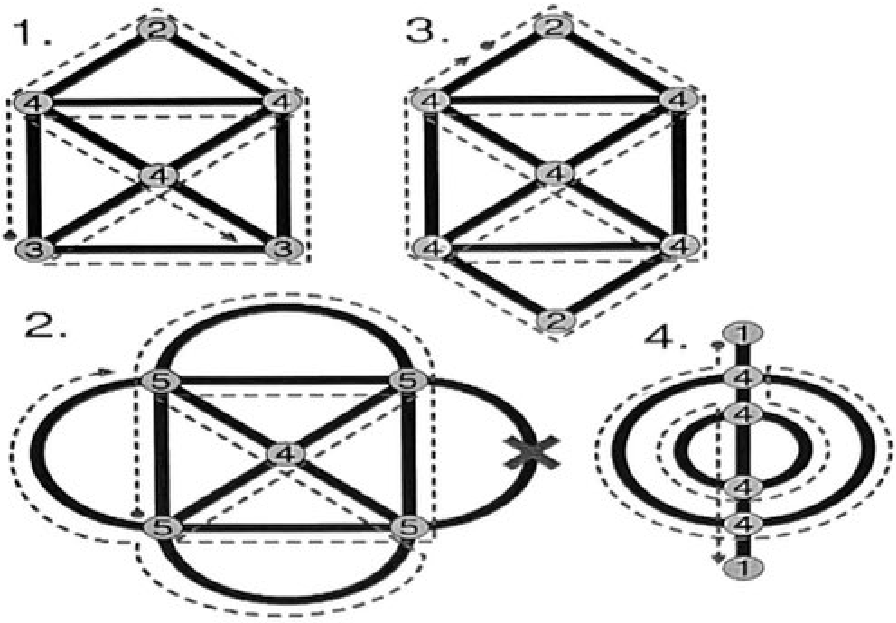
Traversability and Eaulerian Trail

- ❑ Traversability A multigraph is said to be **traversable** if it can be drawn without any breaks in the curve and without repeating any edge.
- ❑ Eulerian trail A gragh G is an eulerian graph if there exists a closed traversable trail, called an eulerian trail.
- ❑ A connected **graph** has an **Euler cycle** if and only if **every** vertex has **even** degree. The term **Eulerian graph** has two common meanings in **graph** theory. One meaning is a **graph** with an **Eulerian** circuit, and the other is a **graph** with **every** vertex of **even** degree. These definitions coincide for connected **graphs**.

Theorem (Euler) A finite connected graph with **two odd** vertices is traversable. A traversable trail may begin at either odd vertex and will end at the other odd vertex.

Corollary Any finite connected graph with two odd vertices is traversable. A traversable trail my begin at either odd vertex and will end at the other odd vertex.

Examples. Which graph of the following are traversable and which are not.



Tree

- A **tree** is an undirected simple graph G that satisfies any of the following equivalent conditions:
- G is connected and has no cycles.
- G has no cycles, and a simple cycle is formed if any edge is added to G.
- G is connected, but is not connected if any single edge is removed from G.
- Any two vertices in G can be connected by a unique simple path.
- G is connected and has $n - 1$ edges.
- G has no simple cycles and has $n - 1$ edges.

A tree with 'n' vertices has 'n-1' edges. If it has one more edge extra than 'n-1', then the extra edge should obviously has to pair up with two vertices which leads to form a cycle. Then, it becomes a cyclic graph which is a violation for the tree graph.

Definitions

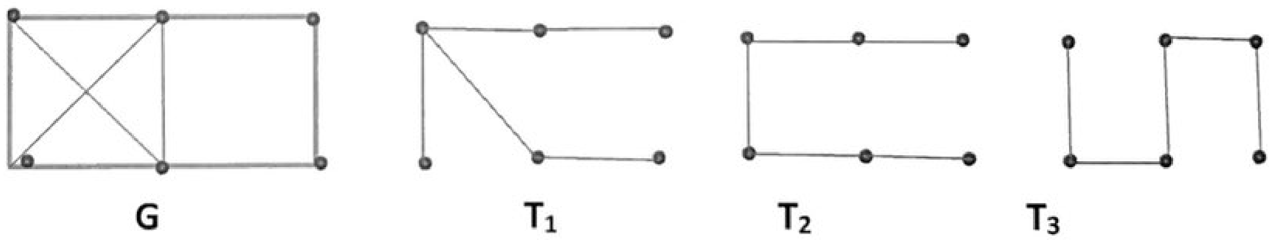
A **tree** is an undirected simple graph *G* that satisfies any of the following equivalent conditions:

- *G* is connected and has no cycles.
- *G* has no cycles, and a simple cycle is formed if any edge is added to *G*.
- *G* is connected, but is not connected if any single edge is removed from *G*.
- Any two vertices in *G* can be connected by a unique simple path.

If *G* has finitely many vertices, say *n* of them, then the above statements are also equivalent to any of the following conditions:

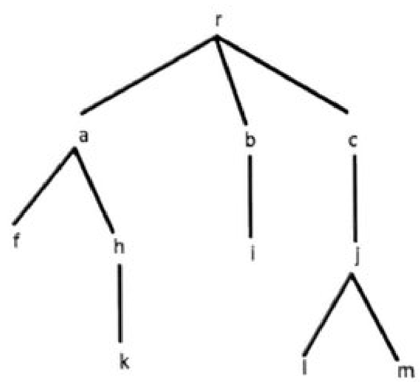
- *G* is connected and has $n - 1$ edges.
- *G* has no simple cycles and has $n - 1$ edges.
  - A **leaf** is a vertex of degree 1. An **internal vertex** is a vertex of degree at least 2.
  - A **rooted tree R** consists of a tree graph together with a designated vertices r called the root of the tree.
  - A **level** is the length of the path from the root r to v.
  - A **branch** is a directed path from a vertex to a leaf.
  - A subgraph T of a graph G is called a **spanning tree** of G if T is a tree and T includes all the vertices of G.

  The following figures shows a graph G and spanning trees T_1, T_2, and T_3 of G.



Example 1

The following figure shows a rooted tree, the root r is at the top of the tree. The tree has five leaves, **f, k, i, l, m**. The level of **a, b,** and c are 1. The level of **f, h,**



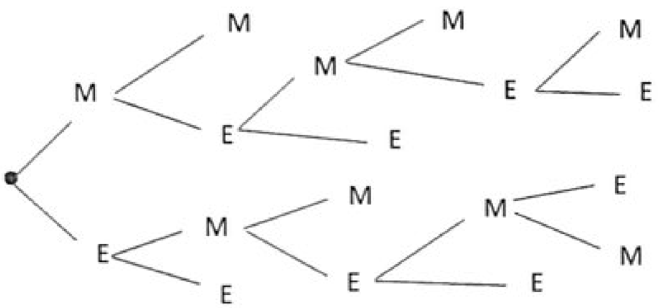
I, and j are 2. The level of **k, l, and m** are 3.

Theorem. Let G be a finite graph with n >1 vertices, then the following are equivalent:

(1) **G** is a tree.
(2) **G** is cycle-free and has n-1 edges.
(3) **G** is connected and has n-1 edges.

Example 2 Suppose two men, Marc and Eric, are playing a tennis tournament such that the first person to win two games in arrow or who wins a total of three games wins the tournament. The ten ways the tournament can occur:

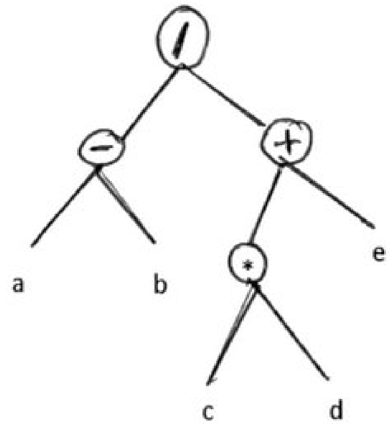
MM, MEMM, MEMEM, MEMEE, MEE, EMM, EMEMM, EMEME, EMEE, EE.



Ordered rooted tree. Any algebraic expression involving binary operations, for example, addition, subtraction, multiplication and division, can be represented by an ordered rooted tree. For example, the following figure represents the arithmetic expression

(a-b) / ((c*d)+e)

1. The variables in the relation represent leaves.
2. The arithmetic operations represent vertices.
3. Last operation represents the root of the tree.



Note. The tree must be ordered since **a-b** and **b-a** yield the same tree but not the same ordered tree.

Give the prefix form: we will symbolize

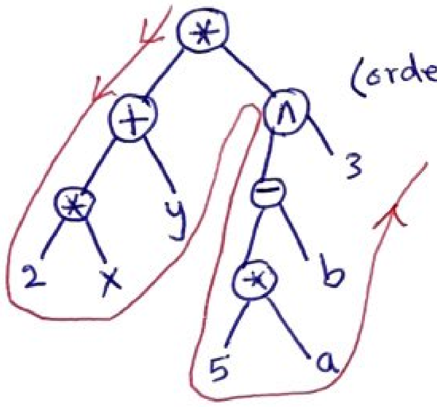
a+b by +ab

a/b by /ab then the relation will be /-ab+*cde

This notation is called Polish notation in prefix form.

Example 3 Consider the algebraic expression $(2x+y)(5a-b)^3$.

(a) Draw the corresponding ordered rooted tree.
(b) Rewrite the expression in prefix polish notation.



Example 4 Evaluate the preorder (prefix) expressions:

+x* yz	$* +xyz$
+2* 34	$* +234$
+212	$* 54$
14	20

Example 5 Give the algebraic expression of each preorder

$+x * yz$ → x+(y* z)

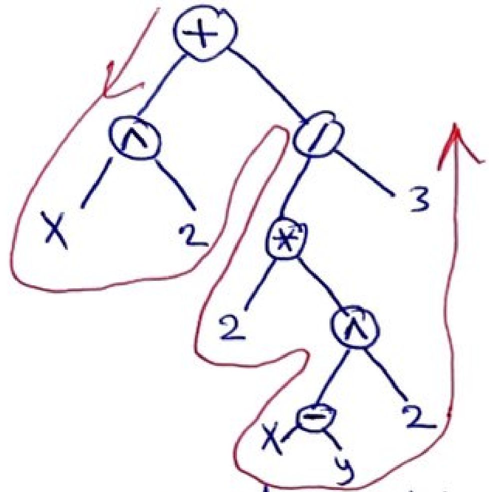
$* +xyz$ → $(x + y) * z$

Example 6 Representing algebraic expression as a tree.

(1) $x^2 + 2 * (x - y)^2/3$

(2) $(a - b) * \frac{3}{2} + (a + b)$

(1)

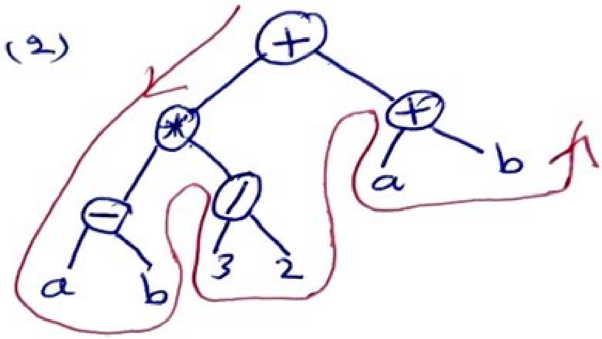


ordered rooted tree

$+ \wedge X\, 2 / * 2 \wedge - X\, y\, 2\, 3$

polish notation

(2)

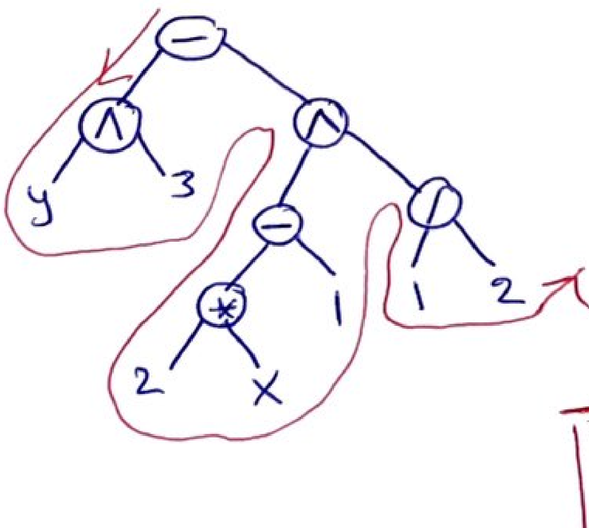


ordered rooted tree

$+ * - a b / 3 2 + a b$

polish notation

(3) $y^3 - \sqrt{2X - 1}$



$- \wedge y 3 \wedge - * 2 X 1 / 1 2$

is the polish notation